Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Marko Bolčič Tavčar

Detekcija in preprečevanje napadov DDOS v okolju manjšega ponudnika internetnih storitev

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

doc. dr. Miha Moškon
MENTOR

Ljubljana, 2018

Tematika naloge:

Kandidat naj v svoji diplomski nalogi preuči nevarnosti napadov DDOS za ponudnike internetnih storitev. Osredotoči naj se na napade, ki temeljijo na poplavljanju UDP. Preuči naj protokole in odprtokodna orodja za detekcijo napadov DDOS in odpravljanje njihovih posledic. Uporabo izbranih orodij naj demonstrira na vzorčnem primeru fiktivnega omrežja manjšega ponudnika internetnih storitev, v katerem naj simulira napad DDOS. Predlagano rešitev za detekcijo in odpravljanje posledic napadov naj kritično ovrednoti na podlagi opravljenih analiz.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani izjavljam, da sem avtor dela, da slednje ne vsebuje materiala, ki bi ga kdorkoli predhodno že objavil ali oddal v obravnavo za pridobitev naziva na univerzi ali drugem visokošolskem zavodu, razen v primerih kjer so navedeni viri.
S svojim podpisom zagotavljam, da:

- sem delo izdelal samostojno pod mentorstvom doc. dr. Mihe Moškona,
- so elektronska oblika dela, naslov (slov., ang.), povzetek (slov., ang.) ter ključne besede (slov., ang.) identični s tiskano obliko in
- soglašam z javno objavo elektronske oblike dela v zbirki "Dela FRI".

— Marko Bolčič Tavčar, Ljubljana, januar 2018.

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Marko Bolčič Tavčar

Detekcija in preprečevanje napadov DDOS v okolju manjšega ponudnika internetnih storitev

POVZETEK

Napadi DDOS v dobi informacijskih oblakov predstavljajo veliko grožnjo za gostovane storitve ter posredno grožnjo izpada dohodkov za podjetja, ki te storitve uporabljajo. Velikokrat so žrtve napadov DDOS prav ponudniki spletnega in aplikativnega gostovanja (ang. *hosting providers*).

V okviru diplomskega dela sem podrobneje opisal proces detekcije in preprečevanja posledic napadov DDOS in predstavil protokole, odprtokodna orodja in metode, ki jih omrežni inženirji vsakodnevno uporabljajo za upravljanje omrežja. Uporabo izbrane rešitve, ki temelji na naštetih protokolih in izbranih odprtokodnih orodij, sem demonstriral na primeru omrežja manjšega ponudnika internetnih storitev. Ta vključuje nadzorni sistem, s katerim sem nadziral ključne elemente omrežja v pričakovanju napada. Simuliral sem volumetrične napade DDOS s poplavljanjem UDP (ang. *UDP flood*) treh velikostnih stopenj. Napade sem zaustavil s pomočjo postopka RTBH. Rezultate sem prikazal v obliki grafov, kjer je razviden začetek napada in zaustavitev z omenjenim postopkom. Iz grafov je bilo razvidno, da se napadeni strežnik med napadom na pakete ICMP skoraj ni več odzival. Usmerjevalniki v omrežju so imeli nadpovprečno obremenjene centralno procesne enote. Po zaustavitvi napada so se kazalniki obremenjenosti omrežja vrnili v prvotno stanje. Predstavljen način preprečevanja napadov se uporablja v okoljih manjših ponudnikov internetnih storitev. Razvoj takih rešitev je odvisen od delovanja odprtokodne skupnosti, delovanje in vzdrževanje pa je odvisno od inženirjev zaposlenih pri ponudniku internetnih storitev. Pri plačljivih rešitvah je, v nasprotju z odportokodnimi, razvoj, delovanje in vzdrževanje garantirano s strani proizvajalca rešitve. Problem tovrstnih rešitev je po drugi strani cenovna nedostopnost za manjše ponudnike.

Ključne besede: DDOS, NetFlow, NFSEN, BGP, RTBH, poplavljanje UDP

University of Ljubljana
Faculty of Computer and Information Science

Marko Bolčič Tavčar

Detection and prevention of DDOS attacks in the environment of a small internet service provider

ABSTRACT

In the age of information clouds DDOS attacks pose a huge threat to hosted services and may cause the loss of revenue for the companies that use these services. Hosting and web application providers are often targets of DDOS attacks.

Herein, I describe the process of detection and prevention of the consequences of DDOS attacks and present the protocols, tools and methods that the network engineers use on a daily basis to manage the communication networks. I have used these protocols and open source tools to establish a solution for DDOS detection and prevention, which is suitable for a small internet service provider (ISP). The solution includes a monitoring system to monitor the key elements of the network in anticipation of the attack. I use the proposed solution on a fictional network in which I simulate DDOS attacks of three different scales using UDP flooding. I demonstrate the attacks mitigation with RTBH method. I analyze the obtained results with the aid of graphs obtained with described tools. The obtained graphs indicate that the attacked server is almost unreachable for ICMP packets during the attack. The routers in the network also have significantly higher CPU utilization than normal. After mitigating the attack, the network load indicators return to their original state. The proposed open source tools are dedicated to the environment of smaller ISPs. Their development depends on the open source community. Moreover, the operation and maintenance of these tools depends on the engineers employed by the ISP. In the case of commercial solutions, the development, operation and maintenance is provided by the vendor of the solution. On the other hand, the problem of such solutions is the price inaccessibility for smaller ISPs.

Key words: DDOS, NetFlow, NFSEN, BGP, RTBH, UDP flooding

ZAHVALA

Najprej bi se zahvalil svojemu mentorju doc. dr. Mihi Moškonu, za pomoč pri usmerjanju skozi diplomsko delo in odzivno pomoč pri popravkih. Posebno bi se zahvalil svoji ženi Mateji, ki me je skozi vsa študijska leta vzpodbujala in mi stala ob strani, da sem pripeljal zadevo do konca. Zahavala tudi moji in ženini družini, ki so verjeli vame.

— Marko Bolčič Tavčar, Ljubljana, januar 2018.

KAZALO

1 Uvod

1.1 Motivacija

Za temo diplomskega dela sem se odločil z namenom, da si pridobim oziroma utrdim dodatno znanje s področja preprečevanja, zaznavanja in odprave napadov DDOS (ang. *distributed denial-of-service attack*). Zaposlen sem v podjetju s področja telekomunikacijskih storitev, v katerem se dnevno srečujemo s problematiko napadov DDOS. Ti v dobi informacijskih oblakov predstavljajo veliko grožnjo za gostovane storitve ter posredno grožnjo izpada dohodkov za podjetja, ki te storitve uporabljajo. Velikokrat so žrtve napadov DDOS prav ponudniki spletnega in aplikativnega gostovanja (ang. *hosting providers*).

V okviru diplomskega dela bom podrobneje opisal metode detekcije in preprečevanja napadov DDOS. Osredotočil se bom na napade, ki temeljijo na poplavljanju UDP (ang. *UDP flooding*). Preučil bom protokole in odprtokodna orodja za detekcijo napadov DDOS in odpravljanje njihovih posledic. Predlagano rešitev za detekcijo in odpravljanje posledic napadov bom kritično ovrednotil na podlagi opravljenih analiz.

1.2 Cilji

V diplomskem delu želim predstaviti orodja in metode, ki jih omrežni inženirji uporabljajo za upravljanje z omrežjem. V današnjih časih se je položaj ponudnikov internetnih storitev, ki jim pravijo tudi varuhi interneta, zelo spremenil. Srečujejo se z izzivi kako ohranjati neprekinjeno delovanje in celovitost omrežja tudi v primeru varnostnih groženj in napadov na uporabnike znotraj njihovih omrežij. V sklopu diplomskega dela bom opisal najbolj razširjene pristope, ki temeljijo na kombinaciji protokolov, ki jih uporabljajo proizvajalci omrežne opreme, in odprtokodnih aplikacijah (ang. *open source applications*).

1.3 Metodologija

V nadaljevanju bom opisal glavne standarde in orodja, ki služijo za detekcijo in preprečevanje DDOS napadov. Med predstavljenimi orodji bom izbral rešitev, ki predstavlja komprosim med odprtokodnostjo orodij in njihovo učinkovtostjo. Uporabo izbrane rešitve bom demonstriral na primeru omrežja, ki vključuje nadzorni sistem, s katerim bom nadziral ključne elemente omrežja v pričakovanju napada. Simuliral bom poskus volumetričnega napada DDOS z različnimi pretoki. Najprej bom z nameščenimi orodji napad zaznal, nato pa napad zajezil oziroma zaustavil. Prikazal bom dva pristopa zaustavitve napada s pomočjo BGP (ang. *Border Gateway Protocol*) protokola. Na koncu bom analiziral rezultate praktičnega poskusa. Opisal bom prednosti in slabosti odprtokodnih rešitev v primerjavi s plačljivimi rešitvami, ki so trenutno na trgu.

1.4 Pregled dela

V drugem poglavju bom podal osnove omrežnih napadov DDOS. V tretjem poglavju bom razložil pristope za detekcijo in odpravo napadov DDOS. V četrtem poglavju bom z uporabo izbranih orodij na primeru demonstriral detekcijo in opravo posledic simuliranih napadov DDOS v fiktivnem omrežju. V petem poglavju bom strnil izsledke opravljenega dela.

2 Omrežni napadi DDOS

V zadnjih dveh letih so se napadi DDOS povečali tako po številu napadov kot tudi po obsežnosti oziroma pasovni širini posameznega napada. Pasovna širina napadov DDOS se dandanes meri v Gbit/s (ang. *gigabits per second*). Največji zabeležen napad je znašal skoraj 800 Gbit/s. Demografsko gledano so najpogosteje napadeni končni uporabniki storitev. Sledijo jim javna uprava, finančne institucije, ponudniki spletnega gostovanja in spletne trgovine. Motivi za napade DDOS so različni. Naštel bom nekaj najbolj aktualnih, sledijo si od najpogostejšega do najmanj pogostega: spletno igranje (ang. *online gaming*), politični ali ideološki motivi, osebne zamere in rivalstva, izsiljevanje, vandalizem, tekmovalnost med podjetji ter spori na socialnih omrežjih [1].

2.1 Najpogostejši tipi napadov DDOS

Napadi DDOS se glede na način svoje izvedbe ločijo na tri glavne sklope. Na kratko jih bom naštel in opisal njihove glavne značilnosti.

- *Zapolnitev tabele stanja povezav* (ang. *TCP state-exhaustion*). Napadalec skuša z napadom DDOS zapolniti tabele stanja omrežnih povezav, ki se uporabljajo na

omrežnih komponentah, kot so požarne pregrade (ang. *firewall*), razporejevalniki obremenitve (ang. *load-balancer*) in aplikacijski strežniki. Ti resursi so pri omrežni opremi omejeni in vezani na strojno opremo. Največkrat je to pomnilnik, kjer se tabele stanja povezav hranijo [1]. Primera takih napadov sta poplavljanje SYN (ang. *SYN Flood*) in ping smrti (ang. *Ping of Death*).

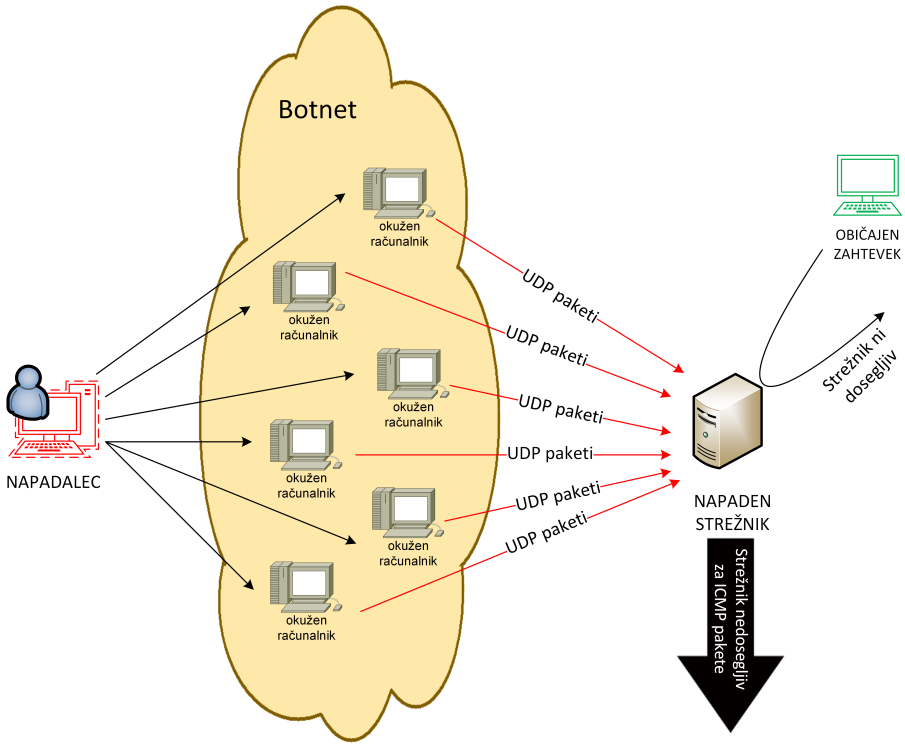
- *Napadi aplikacijske plasti* (ang. *application layer*. Ti napadi so lahko zelo navarni, na samem omrežju pa jih težko zaznamo, saj je dovolj le en napadalec, ki pošilja zelo malo prometa na ranljivo točko aplikacije. Ciljno napadajo slabosti spletnih aplikacij [1]. Primera takih napadov sta poplavljanje HTTP (ang. *HTTP Flood*) in napad na storitve DNS (ang. *Attack on DNS Services*).

- *Volumetrični napadi DDOS*. Napadalec skuša zapolniti razpoložjivo omrežno pasovno širino napadenega omrežja ali storitve. Ponudniki internetnih storitev se srečujejo večinoma s tem tipom napadov, zato se bom v diplomski nalogi posvetil le tovrstnim napadom. Primeri takih napadov so napad z ojačanjem NTP (ang. *NTP Amplification*), napad z ojačanjem DNS (ang. *DNS Amplification*), poplavljanje UDP (ang. *UDP Flood*) in poplavljanje TCP (ang. *TCP Flood*).

2.2 Izvedba napada DDOS

Za napad DDOS velja, da dve ali več naprav pošilja zlonameren promet, ki povzroči, da tarča izgubi dostop do spleta oziroma do dela spleta [2]. Za doseganje teh ciljev napadalec uporabi tako imenovano ojačitev (ang. *amplification*) prometa. Da bi popolnoma ohromil sistem, mora napadalec ustvariti zelo veliko količino prometa, za kar potrebuje veliko naprav. Ojačitev lahko doseže z uporabo omrežja robotskih računalnikov (ang. *botnet*) ali pa z odbojem (ang. *reflection*) prometa preko strežnikov. Pri uporabi omrežja robotskih računalnikov si napadalec največkrat pomaga z zlonamerno programsko kodo, s katero okuži računalnike. Računalniki nato pošiljajo zlonameren promet po napadalčevih ukazih. Pri uporabi odboja za ojačitev napadalec pošlje zahtevek velikemu številu strežnikov. V zahtevek kot izvorni naslov IP vstavi žrtvin naslov IP. Tako strežniki odgovorijo žrtvi in jo posledično poplavijo s prometom [2]. Primer napada je poplavljanje UDP (ang. *UDP flood*), pri katerem začne napadalec s pomočjo okuženih računalnikov pošiljati veliko količino paketov UDP proti žrtvinemu strežniku (glej sliko 2.1). V tem delu se bom osredotočil na tovrstne napade, saj so v okolju ponudnikov

internetnih storitev najpogostejši.



Slika 2.1: Poplavljanje UDP (ang. *UDP flood*) z uporabo t.i. robotskih računalnikov.

3 Pristopi zaznavanja in odpravljanja posledic napadov DDOS

V prvem delu poglavja bom opisal protokole in odprtokodna orodja, ki se uporabljajo pri detekciji napadov DDOS in tudi za odpravo le-teh. Nato bom opisal postopke za zaznavo napadov DDOS in postopke za njihovo odpravo.

3.1 Omrežni protokoli in orodja

3.1.1 Protokol SNMP

Pomembno delo skrbnikov sistemov je zbiranje informacij, ki opisujejo trenutno stanje informacijsko tehnološke (IT) infrastrukture. Obstajajo številna orodja in možnosti za zbiranje tovrstnih informacij. Številna orodja temeljijo na protokolu SNMP (ang. *simple network managment protocol*). Strežniki si lahko z uporabo tega protokola izmenjujejo informacije o svojem trenutnem stanju. Preko SNMP protokola lahko skrbnik sistema določa konfiguracijo strežnikov. Medtem ko je sam protokol zelo preprost, je lahko struktura programov, ki izvajajo SNMP, zelo zapletena [3] .

SNMP je protokol, ki se izvaja na aplikacijskem sloju TCP/IP sklada. Protokol je bil ustvarjen z namenom zbiranja podatkov iz zelo različnih sistemov na dosleden način.

Obstaja več različic protokola SNMP. Najpogosteje uporabljana različica je SNMPv1, vendar je pomanjkljiva z vidika varnosti. Njegova priljubljenost v veliki meri izhaja iz njegove razširjenosti in dolžine obstoja. Dandanes je priporočena uporaba SNMPv3, ki zagotavlja naprednejše varnostne funkcije [3].

Omrežje, ki je nadzirano s pomočjo protokola SNMP, je v glavnem sestavljeno iz naprav, ki imajo vlogo SNMP agentov (ang. *SNMP agents*). Agent je program, ki lahko zbira podatke o strojni opremi, jih organizira v vnaprej določene vnose in se odziva na poizvedbe, ki jih uporablja protokol SNMP. Komponenta, ki agente poizveduje za informacije, se imenuje upravitelj SNMP (ang. *SNMP manager*). Upravitelj SNMP je računalnik, ki je konfiguriran za pošiljanje ukazov SNMP agentom in zbiranje dobljenih informacij. Upravitelj je lahko katerikoli naprava, ki lahko pošlje zahteve za poizvedbe agentom SNMP s pravilnimi poverilnicami. Lahko je del nadzornega sistema, lahko pa ga administrator sistema uporablja za zbiranje določenih podatkov o napravah direktno iz ukazne lupine (npr. temperatura, vlaga itd.). Agenti SNMP naredijo večino dela. Odgovorni so za zbiranje informacij o lokalnem sistemu in za njihovo shranjevanje. Agenti SNMP skrbijo za posodabljanje lokalne baze podatkov, imenovane MIB (ang. *management information base*) [3]. MIB je hierarhična vnaprej določena baza, ki hrani informacije, ki jih je mogoče pregledovati ali nastavljati. Ta je na voljo dobro oblikovanim zahtevam SNMP, ki izvirajo iz gostitelja, z vlogo upravitelja SNMP, ki je identificiran s pravilnimi poverilnicami. Skoraj vsi ukazi, opredeljeni v protokolu SNMP, so namenjeni pošiljanju poizvedb s strani upravitelja. Ti vključujejo *GetRequest*, *GetNextRequest*, *GetBulkRequest*, *SetRequest*, *InformRequest* in *Response*. Poleg tega je upravitelj zasnovan tako, da se odziva na sporočila *Trap* in *Response*, ki jih pošiljajo agenti v določenih okoliščinah. Agenti SNMP se odzivajo na večino ukazov, ki jih določa protokol. Eden od razlogov, da je bil protkol SNMP širše sprejet, je preprostost ukazov. Obstaja zelo malo operacij, vendar so dovolj prožne, da se lahko izvede večina današnjih zahtev. Enote protokola podatkov (ang. *protocol data unit, PDU*) opisujejo točne vrste sporočil preko katerih se pošiljajo ukazi, ki jih dovoljuje protokol [3]:

- *GetRequest* (prodobi vrednost): Pošiljatelj sporočila pošlje sporočilo posredniku, da zahteva vrednost določenega OID (ang. *object identifier*). Na to zahtevo odgovorimo s sporočilom o odzivu, ki se skupaj s podatki pošlje nazaj upravitelju.

- *GetNextRequest* (prodobi naslednjo vrednost): Sporočilo omogoča upravitelju, da v

MIB zahteva naslednji zaporedni predmet. To je način, s katerim lahko prehodimo celotno strukturo MIB.

- *SetRequest* (nastavi vrednost): Upravitelj pošlje agentu sporočilo, da spremeni vrednost, ki jo ima spremenljivka agenta. To lahko uporabimo za nadzor informacij o konfiguraciji ali za spreminjanje stanja oddaljenih gostiteljev. To je edina operacija pisanja, ki jo določa protokol.

- *GetBulkRequest* (prodobi vse vrednosti): Ukaz deluje kot več zaporednih *GetNextRequest* ukazov. Odgovor upravitelju bo vseboval čim več podatkov (v okviru omejitev, ki jih določi zahtevek).

- *Response* (odgovor): To sporočilo, ki ga pošlje agent, se uporablja za pošiljanje zahtevanih podatkov upravitelju. Služi kot prevoz za zahtevane podatke, tudi kot potrdilo o prejemu zahteve. Če zahtevanih podatkov ni mogoče vrniti, odziv vsebuje polja za napake. Za vsako od zgornjih zahtevkov je treba vrniti tako odgovor, kot tudi sporočila.

- Trap (sporočilo o dogodku): Sporočilo o dogodku običajno pošlje agent upravitelju. Dogodki so asinhrona obvestila. Uporabljajo jih predvsem agenti, ki upravitelje seznanijo z dogodki, ki se dogajajo na upravljanih napravah.

- *InformRequest* (potrditev prejema informacije o dogodku): S pošiljanjem obvestila, upravitelj potrdi sprejem sporočila. Če agent ne sprejme tega sporočila, lahko sporočilo o dogodku pošilja še naprej.

Protokol SNMP je šel od svoje vpeljave že skozi številne spremembe odkar je bil uveden. Začetna specifikacija je bila oblikovana z RFC 1065, 1066 in 1067 leta 1988 [3]. Ta različica je še vedno široko podprta, vendar je z vidika varnostni zelo ohlapna, saj uporablja pošiljanje overitve upravitelja v golem tekstu. Delo na različici 2, tj. SNMPv2, se je začelo leta 1993 in ponuja nekaj bistvenih izboljšav napram prejšnjemu standardu [3]. V to različico je bil vključen nov varnostni model, ki temelji na strankah in je namenjen reševanju varnostnih vprašanj povezanih s prejšnjo revizijo. Novi model ni bil zelo priljubljen, ker ga je bilo težko razumeti in izvajati. Leta 1998 je bila objevljena trenutna različica protokola SNMP, tj. različica SNMPv3. Z vidika uporabnika je najpomembnejša sprememba sprejetje uporabniškega varnostnega sistema. Omogoča nastavitev uporabniške prijave z uporabniških imenom in geslom [3].

3.1.2 Orodje NetFlow

NetFlow je orodje, ki je vgrajeno v programski opremi Cisco IOS (ang. *Internetwork Operating System*). NetFlow karakterizira delovanje omrežja in nam omogoča vidljivost omrežja na nivoju kdo, kaj, kdaj, kje in kako pretaka omrežni promet. Tak vpogled v pretok omrežnega prometa nam omogoče, da pravočasno zaznamo težave, ki se oziroma se bodo dogajale znotraj našega omrežja [4].

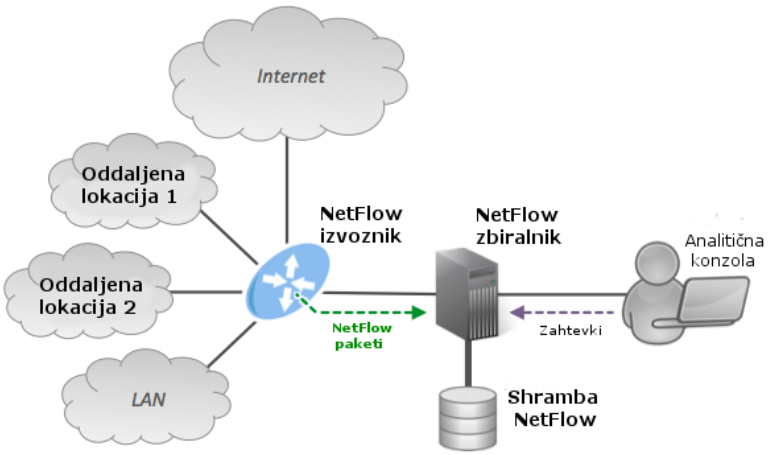
V tradicionalnih IT okoljih se je v preteklosti protokol SNMP uporabljalo izključno za spremljanje zasedenosti omrežja. Čeprav SNMP olajša načrtovanje zmogljivosti pa le malo označuje prometne aplikacije in vzorce, kar je bistveno za razumevanje, kaj se dejansko v omrežju dogaja. Števci paketov in bajtov na vmesnikih so uporabni, vendar je razumevanje, kateri naslovi IP so vir in cilj prometa, in katere aplikacije ustvarjajo promet, neprecenljivo. Sposobnost označevanja omrežnega prometa in razumevanje, kako in kje teče, je ključnega pomena za napovedovanje razpoložljivosi ter zmogljivosti omrežja in odpravljanje težav v omrežju. Spremljanje omrežnih prometnih tokov omogoča natančnejše načrtovanje zmogljivosti omrežja ter ugotavljanje neželenih dogodkov v omrežju, npr. napadov DDOS [4].

Pri analizi NetFlow, je vsak paket, ki je posredovan v usmerjevalniku ali stikalu, pregledan za niz atributov paketa IP. Glavna atributa sta identiteta paketa IP ali prstni odtis paketa in določata, ali je paket edinstven ali podoben drugim paketom. Tradicionalno, pretok IP temelji na nizu od 5 in do 7 atributov IP-paketa:

- izvorni naslov,
- ciljni naslov,
- izvorna vrata,
- ciljna vrata,
- tip protokola na 3. nivoju,
- razred storitve in
- vmesnik usmerjevalnika ali stikala.

Vsi paketi z enakimi izvornimi in ciljnimi naslovi IP, izvornimi in ciljnimi vrati, protokolnim vmesnikom in razredom storitev, so združeni v pretok, nato pa se paketi in bajti

povežejo. Ta metodologija prstnih odtisov ali določanje pretoka je prilagodljiva, ker se veliko informacij o omrežju združi v bazo podatkov NetFlow, ki se imenuje predpomnilnik NetFlow. Za dostop do podatkov NetFlow obstajata dva primarna načina. Prvi je vmesnik ukazne vrstice CLI (ang. *command line interface, CLI*) z ukazi za pregled predpomnilnika NetFlow, drugi pa z uporabo strežnika za poročanje [4]. Najpostejše uporabljen način pregledovanja tokov NetFlow je preko strežnika za poročanje ali t.i. zbiralnika NetFlow (ang. *NetFlow colector*). Zbiralnik NetFlow ima nalogo sestavljanja in razumevanja izvoženih tokov ter združevanja ali zbiranja podatkov za izdelavo poročil, ki se uporabljajo za analizo prometa. Izvoz NetFlow iz usmerjevalnika, za razliko od prejemanja sporočil SNMP, periodično potiska informacije v zbiralnik poročil NetFlow. Na splošno se NetFlow predpomnilnik usmerjevalnika nenehno polni s pretoki. Programska oprema usmerjevalnika v predpomnilniku išče tokove, ki so zaključeni ali pretečeni (ang. *expired*). Ti tokovi se potem izvozijo v zbiralnik NetFlow. Zaključeni tokovi so tisti, kjer se omrežna komunikacija konča (tj. paket vsebuje oznako *TCP FIN*) [4]. Obstajajo različni formati za izvoz, ki se običajno imenujejo izvozne različice. Izvozne različice so dobro dokumentirane oblike, vključno z različico 5, 7 in 9. Najpogostejša oblika, ki se uporablja, je NetFlow izvozna različica 5, različica 9 pa je najnovejša oblika in ima nekaj prednosti za ključne tehnologije, kot so varnost, analiza prometa in *multicast* promet. Lokacija, kjer je nameščen NetFlow, je odvisna od lokacije rešitve za poročanje in topologije omrežja glej sliko 3.1. Če je strežnik za zbiranje poročil centralno nameščen, je uporaba NetFlow blizu strežnika za poročanje zbiratelja optimalna. Pri izvozu v zbirni strežnik se uporablja približno 1 do 5 % pasovne širine preusmerjenega prometa.



Slika 3.1: NetFlow arhitekura. [5]

Obstaja veliko število zbiratelnikov NetFlow. Lahko so brezplačni (ang. *freeware*) ali komercialni. NetFlow je pomembna tehnologija, ki je na voljo v boljših usmerjevalnikih, in pomaga prepoznati, kako se uporabljajo omrežna sredstva in kakšno je obnašanje omrežja.

3.1.3 Protokol BGP

BGP (ang. *Border Gateway Protocol*) je standardiziran protokol med robnimi usmerjevalniki, namenjen izmenjavi informacij o poti in dosegljivosti med avtonomnimi sistemi na internetu. BGP pomaga pri sprejemanju odločitev o usmerjanju na podlagi poti, politike usmerjanja ali nastavitev pravil, ki jih konfigurira skrbnik omrežja [6].

Neodvisna usmerjevalna domena, ki skoraj vedno pomeni omrežje ponudnika storitev, se v BGP svetu imenuje avtonomni sistem (AS). BGP se večinoma uporablja za usmerjevalne odločitve na robnih usmerjevalnikih ponudnika storitev, torej za odločitve o izbiri ponudnika preko katerega bo avtonomni sitem dostopal do interneta. V večjih omrežjih se lahko BGP uporablja kot interni usmerevalni protokol. Takrat ga imenujemo iBGP oziroma interni BGP protokol [7].

Ostali protokoli za usmerjanje se nanašajo izključno na iskanje optimalne poti do vseh znanih destinacij. BGP je že po svoji strukturi kompleksenejše zasnovan, saj mora upoštevati politike usmeranja, ki se vodijo med ponudniki storitev internet. Če želi BGP pomagati omrežnim operaterjem izvajati te politike, nosi v sebi veliko število atributov.

Glavni BGP atributi [7] so:

- Pot AS (ang. *AS path*), je redosled poti čez katere AS mora paket potovati da doseže cilj.
- Lokalna preferenca (ang. *local preference*) nam daje informacijo o prednostni poti za izhod iz AS, da bi dosegli določeno omrežje.
- Atribut večizhodni diskriminator (ang. *multi-exit discriminator*) daje sosednjim ponudnikom internetnih storitev možnost, da izberejo eno medomrežno povezovalno točko pred drugo.
- Atribut skupnost (ang. *community*) je niz generičnih oznak, ki jih lahko uporabimo za signaliziranje različnih administrativnih pravilnikov med BGP usmerjevalniki.

3.1.4 Orodje RRDtool

RRDtool (ang. *round-robin database tool*) je grafično orodje, ki prebere zbrane zbirke podatkov in jih prikaže v obliki časovnega grafa [8] . RRDtool povezuje časovne vrste podatkov, kot so pasovna širina v omrežjih, temperature in obremenitev CPU (ang. *central processing unit*). Podatki so shranjeni v podatkovni zbirki tipa *round-robin* (krožni medpomnilnik), ki omogoča, da ostane velikost RRD datotek s časom nespremenjena. Baza podatkov bo vedno imela enako količino podatkovnih točk v celotni življenjski dobi. Ko pridejo novi podatki, je odstranjen najstarejši niz podatkov po principu FIFO (ang. *first in, first out*). Orodje uporablja mnogo priljubljenih grafičnih programov, kot so Cacti, SmokePing, MRTG, Nagios, Zenoss, collectd, Astaro, Nmon in drugi [8].

3.1.5 Orodje Cacti

Cacti je spletni programski paket za prikaz grafov, ki temeljijo na RRD (ang. *round-robin database*) formatu. Shranjuje vse potrebne informacije za ustvarjanje grafov in jih posodablja v podatkovni bazi MySQL. V bazi podatkov je sposoben hraniti slike, vire podatkov in RRD arhive. Hkrati lahko zbira in obdeluje nove podatke. Prav tako se lahko ustvarijo viri podatkov, ki ustrezajo dejanskim podatkom na grafu. Če bi uporabnik na primer želel grafični prikaz odziva ping do nadzorovanega elementa v omrežju, lahko ustvarimo vir podatkov z uporabo skripte, ki pošilja zahtevke ping nadzorovenemu elementu in vrne vrednost odziva v milisekundah. Zbrani podatki se zapišejo v

datoteko RRD. Tako ustvarjen vir, se samodejno vzdržuje v 5-minutnih intervalih. Ko je določen eden ali več virov podatkov, lahko z uporabo podatkov ustvarimo graf z pomočjo orodja RRDTool [9]. Zaradi velikega številnih vgrajenih funkcij znotraj portala Cacti, je bilo razvito uporabniško orodje za upravljanje uporabniških pravic, tako da lahko dodajamo uporabnike in jim določamo pravice za dostop do področij portala. To nam omogoča, da ustvarimo nekaj uporabnikov, ki lahko spremenijo parametre grafov, drugi pa si lahko grafe le ogledujejo. V orodju Cacti lahko uporabljamo tudi predloge, ki jih lahko povežemo z velikim številom podatkovnih virov in grafikonov. To omogoča oblikovanje enotne grafične podobe ali predloge podatkovnega vira, ki določa katerikoli graf ali vir podatkov, povezan z njim. Predloge nadzorovenega elementa nam omogočajo, da določimo zmožnosti nadzorovenga elemeta, tako da lahko Cacti zbere pravilne informacije ob vnosu novega elementa omrežja [9].

3.1.6 Orodje NFSEN in NFDump

NFSEN je spletno orodje za prikaz podatkov *NetFlow* pridobljenih z orodjem NFDump. Orodje NFDump je nastalo zaradi potrebe po analizi podatkov *NetFlow* izven ukazne lupine samega usmerjevalnika. NFDump hrani podatke v časovno rezanih datotekakah (ang. *time sliced files*), kar mu omogoča tudi zgodovinski pregled prometnih tokov. Zna hitro analizirati velike količine podatkov *NetFlow* tudi na skromnejšem strežniku. Da bi lahko podatke NFDump uporabniku prikazali v obliki grafov, so v ta namen razvili orodje NFSEN. Vgrajen ima tudi modul za obveščanje, ki nas lahko preko elektronske pošte obvesti o prekoračitvi vnaprej nastavljenega praga, ki je določen z številom paketov v sekundi ali s pretokom v Mbit/s [10].

3.2 Postopek detekcije in zaustavitve volumetričnega napada DDOS

V tem poglavju bom opisal kako lahko s pomočjo dveh izbranih orodij detektiramo napad DDOS. Nato bom opisal še dve tehniki zaustavitve napada na robnih točkah omrežja.

3.2.1 Detekcija napada s protokolom SNMP

Skoraj vsak ponudnik intenetnih storitev uporablja eno od orodij, ki preko protokola SNMP, nadzira pretoke vhodno-izhodnih vmesnikov omrežnih stikal in usmerevalnikov. Tipično se podatki SNMP zbirajo z naprav vsakih 5 minut in se vpisujejo v datoteke RRD. To pomeni, da imamo podatek o povprečnem prometu na vmesniku za vsakih

5 minut. Določena orodja imajo na voljo modul, kjer lahko nastavimo prag prometa za določen vmesnik. V kolikor je prag presežen nas orodje o dogodku obvesti preko elektronske pošte. Preko protokola SNMP lahko tako detektiramo povečanje prometa na določenih vmesnikih, ne vemo pa kdo ali kaj to povečanje povzroča. V sled temu anomalijo težko odpravimo.

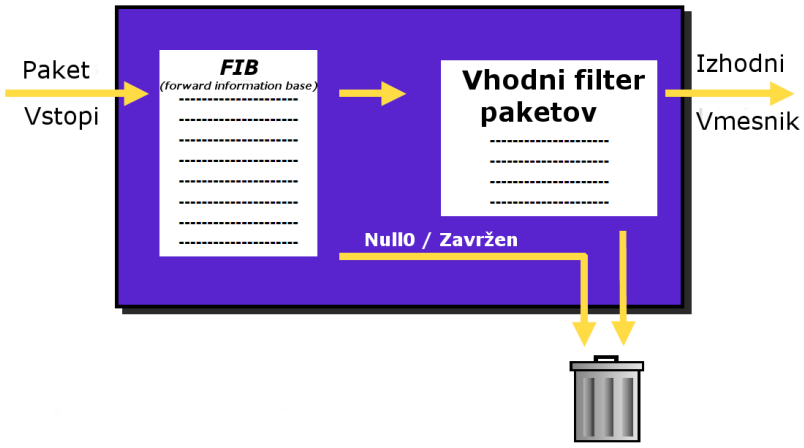
3.2.2 Detekcija napada z orodjem NetFlow

Pri NetFlow detekciji lahko preko orodja NFSEN dobimo opozorilo, da imamo anomalijo v omrežju glede na nastavljene prage. S podrobno analizo podatkov NFDump ugotovimo, da iz interneta prihaja do napada DDOS. Določimo lahko tudi tarčo napada. Vidimo kakšen je napad v smislu pasovne širine in števila poslanih paketov na sekundo. V tem primeru nimamo več dvoma, da gre za napad DDOS in lahko postopamo naprej s spodaj opisanimi metodami za zaustavitev napada DDOS.

3.2.3 Omejitev posledic napada z usmerjanjem *Blackhole*

Opisal bom kako se usmerjanje oziroma filtriranje *Blackhole* uporablja v praksi. Večina sodobnih usmerjevalnikov ima poseben psevdovmesnik, ki se običajno imenuje *Null0*. Ta je vedno aktiven in nikoli ne more posredovati ali sprejemati prometa. Paket, ki je usmerjen na Null0, bo zavržen s strani usmerjevalnika, zato se lahko ta vmesnik uporabi za zavračanje neželenega prometa. Ta metoda je bolj učinkovita kot uporaba seznamov za nadzor dostopa (ang. *access lists*). Slika 3.2 prikazuje kako potujejo zavrnjeni omrežni paketi v usmerjevalnikovi logiki. Metoda ne prinaša nobenih dodatnih stroškov, saj uporablja visoko optimiziran usmerjevalni postopek znotraj usmerjevalnika. Spodaj je prikazan primer CLI ukaza na Cisco usmerejvalniku, ki usmerni omrežje 172.16.10.100/32 v psevdovesnik *Null0*, tj. usmerjevalnik bo zavrgel vse pakete kjer je cilj navedeno omrežje [11].

R9(config)# ip route 172.16.10.100 255.255.255.255 Null0



Slika 3.2: Usmerjanje *Blackhole*. [12]

3.2.4 Omejitev posledic napada s postopkom *RTBH*

Prestavljajmo si, da je ena stranka v našem omrežju tarča velikega napada DDOS. Napad je tako velik, da lahko povzroči zasičenje notranjih medomrežnih povezav. Zato moramo z metodo usmerjanja *Blackhole* ukrepati hitro ter tako zaustaviti napad na robnih usmerjevalnikih (ang. *border routers*), po možnosti na vseh hkrati. V primeru večjega omrežja je lahko teh usmerjevalnikov veliko in bi samo z metodo usmerjanja *Blackhole* težko dosegli hitro konvergenco blokiranja napada.

Bolj učinkovita metoda blokiranje se imenuje RTBH (ang. *remotely triggered blackhole*). Metoda se poslužuje BGP protokola ozirom iBGP. V omrežje iBGP dodamo usmerjevalnik, ki ga imenujemo sprožilni usmerjevalnik (ang. *trigger router*). Na kratko bomo opisal postopek aktivacije RTBH na Cisco usmerejevalniku.

Najprej na robnih usremejevalnikih nastavimo statično usmerjanje privatnega IP naslova na vmesnik *Null0*. Naslov 192.0.2.1/32 spada v segment *TEST-NET* po standardu RFC3300 [13]. Večina navodil za izvedbo RTBH uporablja ta naslov. Spodaj je prikazan ukaz za Cisco usmerevalnik [11].

R1(config)# ip route 192.0.2.1 255.255.255.255 Null0

Na sprožilnem usmerjevalniku nastavimo usmerjevalno razporejanje (ang. *route map*), ki pove usmerjevalniku, da se lahko po iBGP protokolu oglašuje oziroma redistribuira samo določeno statično usmerjanje z določeno usmerjevalno labelo (ang. *route tag)*. Spodaj je naveden niz ukazov za Cisco usmerjevalnik [11].

R9(config)# route-map RTBH

R9(config-route-map)# match tag 666

R9(config-route-map)# set ip next-hop 192.0.2.1

R9(config-route-map)# set origin igp

R9(config-route-map)# set community no-export

To je ključna komponenta za metodo RTBH. Vsaka pot, oglaševana na robnih usmerevalnikih z naslednjim skokom (ang. *next-hop*) 192.0.2.1, bo sprožila rekurzijo na statično usmerjanje *Null0*, ki smo jo izvedli v predhodni konfiguraciji. Sledilo bo odmetavanje prometa proti tej poti. Spodaj je prikazan primer ukaza statičnega usmerjanja z RTBH labelo [11].

R9(config)# ip route 172.16.10.100 255.255.255.255 Null0 tag 666

Promet namenjen proti IP naslovu 172.16.10.100 bo zavržen na robnih usmerevalnikih. S tem smo zajezili napad in obvarovali omrežno infrastrukturo.

BGP Flowspec je razširitev protokola BGP po RFC 7674 [14], ki omogoča bolj granularni pristop zavračanja neželenega prometa proti zunanjem omrežju ali iz njega. Omogoča dinamično namestitev režima delovanja na robnih usmerjevalnikih kot naprimer [15]:

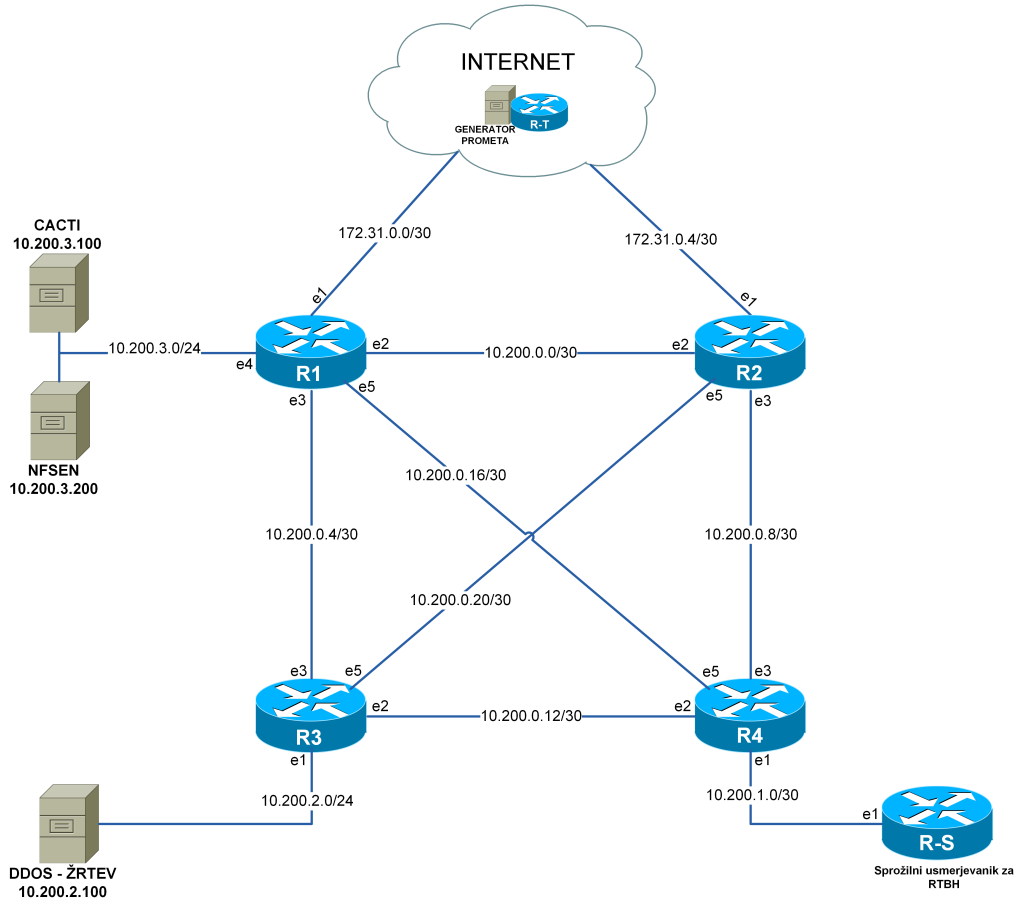
- odvrzi promet,
- usmeri promet v posebno omrežje za nadaljnjo analizo,
- dovoli promet vendar ga omeji na določeno stopnjo.

4 Primer detekcije in odprave posledic napada DDOS

V namen predstavitve poteka napada DDOS in njihove detekcije ter odprave, sem zasnoval fiktiven in poenostavljen primer omrežja ponudnika internetnih storitev. Uvodoma bom opisal glavne gradnike in strukturo omrežja. Nato bom predstavil potek simulacije, detekcije in odprave napada DDOS, torej kateri so bili vhodni podatki v simulacijo, kako sem rezultate shranil za izvedbo analize napada in kako sem napad odpravil. V zaključku bom pridobljene rezultate komentiral.

4.1 Opis testnega omrežja

Slika 4.1 prikazuje demostracijsko omrežje, ki predstalja poenostavljeno omrežje ponudnika internetnih storitev.



Slika 4.1: Demostracijsko omrežje fiktivnega ponudnika internetnih storitev.

Omrežje je zgrajeno v virtualnem okolju VMWARE [16]. Osrednji del je sestavljen iz štirih usmerjevalnkov, ki so med seboj polno povezani (ang. *full mesh*). Poimenovani so od R1 do R4. Peti usmerjevalnik je povezan na R4 in nosi ime RS, kar naj bi pomenilo sprožilni usmerjevalnik. Ta usmerjevalnik bo uporabljen za oglaševanje napadenega omrežja ali naprave. Usmerjevalnika R1 in R2 sta preko omrežnega vmesnika e1 povezana na internet. Za simulacijo interentnega dostopa sem namestil še en usmerjevalnik RT, preko katerega bom simuliral pošiljanje prometa napada DDOS. Pasovna širina vseh povezav v omrežju je 1 Gbit/s. Usmerjevalniki poganjajo Mikrotik RouterOS programsko opremo verzija 6.41 [17]. V tabeli 4.1 je prikazano IP naslavljanje omrežnih vmesnikov na usmerevalnikih v prefiksni notaciji (ang. *prefix notation*).

Tabela 4.1: IP naslovi usmerjevalnikov v testnem omrežju.

USMERJE-VALNIK	**OMREŽNI VMESNIK**				
	e1	e2	e3	e4	e5
R1	172.31.0.2/30	10.200.0.1/30	10.200.0.5/30	10.200.3.1/24	10.200.0.17/30
R2	172.31.0.6/30	10.200.0.2/30	10.200.0.9/30		10.200.0.21/30
R3	10.200.2.1/24	10.200.0.13/30	10.200.0.6/30		10.200.0.22/30
R4	10.200.1.1/30	10.200.0.14/30	10.200.0.10/30		10.200.0.18/30
RS	10.200.1.2/30				
RT			172.31.0.1/30	172.31.0.5/30	

Za usmerjanje znotraj omrežja sem izbral OSPF protokol (ang. *Open Shortest Path First*). Protokol poskrbi, da so vse destinacije znotraj omrežja vedno dosegljive preko neposrednih sosedskih povezav. Na usmeravalnikih od R1 do R4 in RS sem vzpostavil še BGP usmerjanje z AS številka 65530. BGP seje sem aktiviral med vsakim neposredno povezanim sosedom. Torej imam aktivne seje med R1-R2, R1-R3, R1-R4, R2-R4, R2-R3 in R4-RS. Na usmerjevalnikih R1 in R2 sem vključil BGP filter, ki dovoljuje sprejem *BlackHole* usmerjene poti z labelo skupnosti 666 na BGP seji z usmerjevenikom R4. Spodaj je prikazan ukaz za izgradnjo filtra in dodajanje le-tega na BGP sejo za usmerejevalnik R1.

[admin@R1] >routing filter add action=accept bgp-communities=65530:666
chain=bgp-in set-type=blackhole
[admin@R1] >routing bgp peer add in-filter=bgp-in name=R4
remote-address=10.200.0.18 remote-as=65530 update-source=ether5

V omrežje sem umestil tri strežnike. Strežnik povezan na usmerejevalnik R3 ima vlogo žrtve in sem ga imenoval ŽRTEV. V simulaciji bomo nanj usmerili napad DDOS s poplavljanjem UDP. Strežnika povezana na usmerjevalnik R1 poganjata orodji za nadzor omrežja Cacti in NFSEN. Na strežnike sem namestil operacijski sistem CentOS 7. Orodje Cacti sem namestil po navodilih iz vira [18], NFSEN pa po viru [19]. V tabeli 4.2 so navedeni IP naslovi strežnikov.

Tabela 4.2: Naslovi strežnikov v testnem omrežju.

STREŽNIK	**IP NASLOV**
CACTI	10.200.3.100/24
NFSEN	10.200.3.200/24
ŽRTEV	10.200.2.200/24

Za potrebe SNMP nadzora imajo vsi elementi v omrežju aktivirane SNMP funkcije in nastavljeno privzeto *community* nastavitev na *public*. Spodaj je primer ukaza na usmerjevalniku R1, za zgoraj navedene nastavitve.

[admin@R1] >snmp community set name=public;snmp set enable=yes;

Usmerjevalnika R1 in R2 imata nameščeno konfiguracijo za izvoz NetFlow podatkov proti NFSEN strežniku. Spodaj je zapisan primer ukaza za usmerjevalnik R1.

[admin@R1] >ip traffic-flow set active-flow-timeout=1m cache-entries=128k enabled=yes
[admin@R1] >ip traffic-flow target add dst-address=10.200.3.200 src-address=10.200.0.1

V orodju Cacti sem po navodilih iz vira [18] vnesel usmerjevalnike in strežnik ŽRTEV v seznam nadzorovanih naprav. Za usmerjevalnike sem nastavil grafiranje omrežnih vmesnikov in omremenitve procesorja. Dodal sem tudi graf, ki spremlja odzivnost elementov v omrežju v odvisnoti od poslanih ICMP (ang. *Internet Control Message Protocol*) paketov.

4.2 Opis simulacije napada DDOS

Za generiranje napada DDOS sem uporabil modul *Traffic generator* [20], ki je del RouterOS programske opreme usmerjevalnika RT. S pomočjo modula za generiranje prometa sem simuliral napad na strežnik ŽRTEV. V tabeli 4.3 so prikazani parametri prometnih tokov, ki sej jih vnesel v generator prometa na usmerjevalniku RT.

Tabela 4.3: Podatki o prometnih tokovih v testnem omrežju.

# toka	izvorni IP naslov	ponorni IP naslov	IP prehoda	protokol	izvorna vrata	ponorna vrata
1	10.236.120.41	10.200.2.200	172.31.0.2	UDP	16122	53
2	10.84.138.200	10.200.2.200	172.31.0.2	UDP	47255	22
3	10.47.10.134	10.200.2.200	172.31.0.2	UDP	41049	5060
4	10.149.162.219	10.200.2.200	172.31.0.2	UDP	42588	110
5	10.220.73.88	10.200.2.200	172.31.0.2	UDP	21046	161
6	10.152.96.117	10.200.2.200	172.31.0.2	UDP	44142	25
7	10.123.211.203	10.200.2.200	172.31.0.2	UDP	11002	320
8	10.191.101.5	10.200.2.200	172.31.0.6	UDP	23456	3389
9	10.42.187.65	10.200.2.200	172.31.0.6	UDP	12378	37
10	10.233.230.111	10.200.2.200	172.31.0.6	UDP	34256	21
11	10.34.252.99	10.200.2.200	172.31.0.6	UDP	54367	69
12	10.195.173.245	10.200.2.200	172.31.0.6	UDP	22341	465
13	10.220.201.247	10.200.2.200	172.31.0.6	UDP	61236	514
14	10.163.226.214	10.200.2.200	172.31.0.6	UDP	5890	548

V vsaki iteraciji simulacije sem povečeval količino prometa vsakega toka za faktor dva (prikazabno v tabeli 4.4). Generiranje prometa sem razdelil enakomerno proti usmerjevalnikoma R1 in R2.

Tabela 4.4: Pretok prometa v posameznih simulacijah.

# toka	simulacije št. 1 pretok v Mbit/s	simulacije št. 2 pretok v Mbit/s	simulacije št. 3 pretok v Mbit/s
1	2	4	8
2	3	6	12
3	5	10	20
4	5	10	20
5	4	8	16
6	8	16	32
7	3	6	12
8	8	16	32
9	2	4	8
10	7	14	28
11	5	10	20
12	2	4	8
13	5	10	20
14	1	2	4

Pri prvi in drugi simulaciji sem generiral promet za obdobje 45 minut. Pri zadnji simulaciji sem moral skrajšati čas na 20 minut, ker so se usmerjevalniki začeli naključno resetirati že po 30 minutah, predvidevam da zaradi prevelike količine prometa. V pomoč mi je bil razporejevalnik opravil (ang. *task scheduler*) usmerjevalnika RT. Z njim sem določil točen čas začetka in konca generiranja prometa. Spodaj je prikazana koda za aktivacijo predhodno opisanih prometnih tokov.

[admin@RT] >tool traffic-generator start stream=1,2,3,4,5,6,7,8,9,10,11,12,13,14;

Prva simulacija je generirala skupaj 60 Mbit/s, druga 120 Mbit/s, tretja pa 240 Mbit/s UDP prometa. Na usmerjevalniku R3 sem s pomočjo funkcije *queue* [21] omejil maksimalni pretok do strežnika ŽRTEV na 50 Mbit/s. Spodaj je prikazan ukaz za omejitev pretoka pri omenjenemu strežiku.

[admin@R3] >queue simple add max-limit=50M/50M name=queue1
target=10.200.2.200/32;

Pri prvi in drugi simulaciji sem po 30 minutah izvedel metodo RTBH na usmerjevalniku RS. Pri tretji sem moral izvesti RTBH že po šestih minutah simulacije, saj so se usmerjevalniki zaradi prevelike količine prometa predčasno nehali odzivati ali pa so se naključno resetirali. Spodaj je prikazan ukaz dodajanja usmerjanja za RTBH na usmerjevalniku RS.

[admin@RS] >ip route add bgp-communities=65530:666 distance=1
dst-address=10.200.2.200/32 type=blackhole;

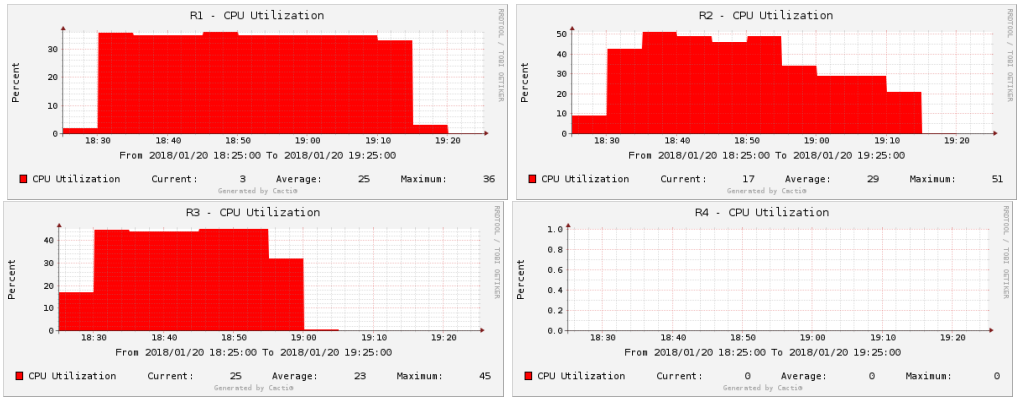
4.3 Analiza rezultatov

V analizi sem se osredotočil le na rezulrate druge simulacije, saj so ti najbolj reprezentatitvni. Rezultati prve simulacije so zelo podobni in so le manjši po skali. Rezultatov tretje simulacije nisem predstavil, ker je nisem uspel izvesti v enaki meri kot prvo in drugo simulacijo, saj je testno okolje zdržalo prometno breme le omejeno časovno obdobje. V sled temu so rezultati tretje simulacije nereprezentativni.

Po pričakovanjih je promet do strežnika ŽRTEV v polovici potekal čez usmerjevalnik R1 in v polovici čez usmerjevalnik R2. Preko usmerjevalnika R3 je promet prišel v omrežje napadenega strežnika. Na sliki 4.2 je prikazan promet usmerjevalnikov R1, R2 in R3, in sicer na povezavah, kjer je potekal napad v drugi simulaciji. Napad se je začel ob 18:30. Na zgornjih dveh grafih se lepo vidi vhodni promet usmerjevalnika R1 in R2 na omrežnem vmesniku e1. Osrednja dva grafa prikazujeta odhodni napad iz usmerjevalnkika R1 (vmesnik e3) in usmerjevalnika R2 (vmesnik e5). Graf prometa je na teh slikah prikazan z modro črto saj gre za izhodni promet, kot to opisuje legenda pod grafom. Spodnja dva grafa prikazujeta vhod napada na R3 na vmesnikih e3 in e5. Zadnji graf prikazuje izhod napada na vmesniku e1 proti strežniku ŽRTEV. Ponovno je prikazan z modro črto saj gre za izhodni promet, velikostnega reda 50Mbit/s kot določuje pravilo omejitve pretoka na usmerejvalniku R3. Ob 19:00 se izvede metoda RTBH, kar je razvidno na grafih usmerjevalnikov R1 in R2 na vmesnikih e3 in e5. Prav tako je to razvidno na grafih usmerjevalnika R3 na vmesnikih e1, e3 in e5. Napad je še vedno prisoten na vhodu v omrežje do 19:15 kar je razvidno na zgornjih dveh grafih. Napad se zaključi ob 19:15. Takrat je viden tudi padec prometa na R1 in R2 na vmesnikih e1.

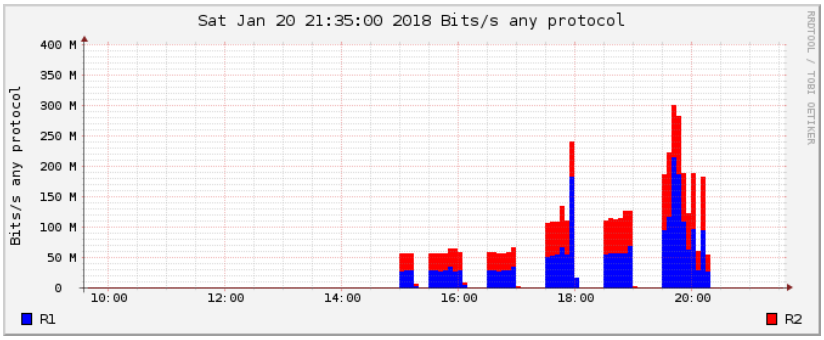
Slika 4.2: Vizualizacija rezultatov druge simulacije. Napad se je začel ob 18:30 in končal ob 19:15. Metoda RTBH se je izvedla ob času 19:00.

Na slikah 4.3 so prikazane obremenitve procesorskih enot usmerjevalnikov. Najbolj obremenjena je enota usmerjevalnika R3, saj sprejema promet iz R1 in R2 ter ga pošilja naprej napadenemu strežniku na vmesnik e1. Slike ostalih dveh simulacij sta podobnih oblik, razlika je le v večjem ali manjšem prometu in večji ali manjši utilizaciji procesorskih enot.



Slika 4.3: Slike obremenitve procesorskih enot usmerjevalnikov.

Na sliki 4.4 je prikazan graf pridobljen z orodjem NFSEN tekom vseh treh simulacij. Razvidna je količina prometa, ki je potekala preko R1 in R2 v času simulacij.



Slika 4.4: Analiza obremenitve usmerjevalnikov R1 in R2 z orodjem NFSEN. Prva simulacija se je izvajala v časih med 14:45 do 17:30 (prvi trije stolpci na grafu), druga simulacija se je izvajala med 17:30 in 19:30 (četrti in peti stolpec na grafu), tretja simulacija pa v času med 19:30 in 20:30 (zadji stolpec na grafu). Metoda RTBH je bila sprožena ob času 15:15, 16:00, 17:00, 18:00, 19:00 in 20:16

Več informacij dobimo če gremo globlje in naredimo NetFlow analizo podatkov med napadom. Spodaj je naveden ukaz za orodje NFDump za časovno obdobje napada druge simulacije.

nfdump -M /opt/nfsen/profiles-data/live/R2:R1 -T -R 2018/01/20/nfcapd.201801201830: 2018/01/20/nfcapd.201801201900 -n 20 -s ip/bps

Ukaz požene analizo podatkov NetFlow za prvo datoteko z imenom nfcapd.201801201830

in nato za vse vmesne datoteke po vrstnem redu vključno z zadnjo datoteko z imenom nfcapd.201801201900. Razvrsti tokove od najbolj obremenjenega v bps (ang. *bits per second*) do najmanj obremenjenega. Na sliki 4.5 je prikazan primer izpisa za predhodno prikazan ukaz.

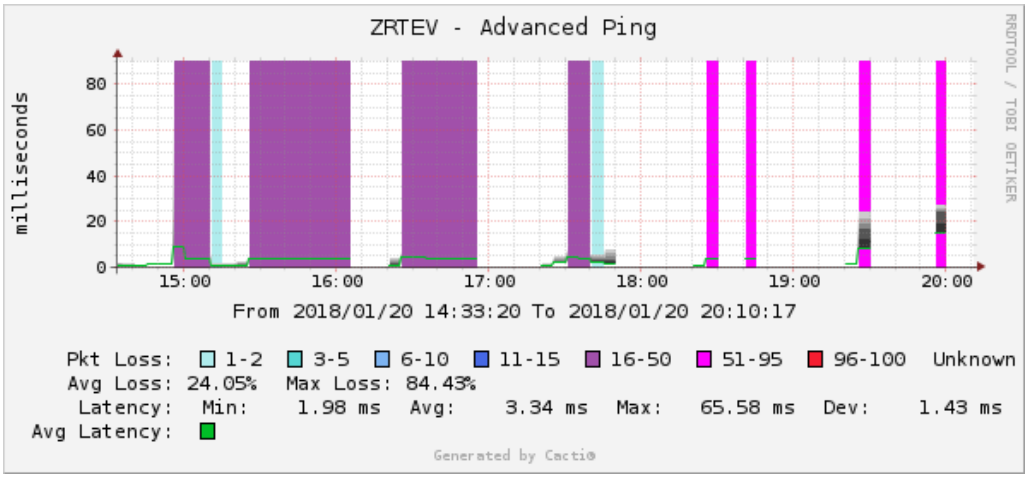
Top 20 IP Addr ordered by bps:

Date first seen	Duration	Proto	IP Addr	Flows(%)	Packets(%)	Bytes(%)	pps	bps	bpp
2018-01-20 18:29:38.320	1798.800	any	10.200.2.200	760(13.8)	17.7 M(99.3)	26.6 G(99.8)	9857	118.2 M	1498
2018-01-20 18:29:38.320	1796.470	any	10.152.96.117	51(0.9)	2.4 M(13.3)	3.5 G(13.3)	1316	15.8 M	1499
2018-01-20 18:29:38.330	1798.760	any	10.191.101.5	52(0.9)	2.4 M(13.2)	3.5 G(13.3)	1310	15.7 M	1499
2018-01-20 18:29:38.320	1798.770	any	10.233.230.111	52(0.9)	2.1 M(11.6)	3.1 G(11.6)	1146	13.8 M	1499
2018-01-20 18:29:38.320	1793.470	any	10.149.162.219	50(0.9)	1.5 M(8.3)	2.2 G(8.3)	824	9.9 M	1498
2018-01-20 18:29:38.320	1798.770	any	10.220.201.247	52(0.9)	1.5 M(8.3)	2.2 G(8.3)	822	9.9 M	1498
2018-01-20 18:29:38.320	1796.480	any	10.47.10.134	50(0.9)	1.5 M(8.3)	2.2 G(8.3)	822	9.9 M	1498
2018-01-20 18:29:38.320	1798.770	any	10.34.252.99	52(0.9)	1.5 M(8.3)	2.2 G(8.3)	821	9.9 M	1498
2018-01-20 18:29:38.320	1792.200	any	10.220.73.88	25(0.5)	1.2 M(6.6)	1.8 G(6.7)	659	7.9 M	1500
2018-01-20 18:29:38.320	1797.480	any	10.123.211.203	51(0.9)	889018(5.0)	1.3 G(5.0)	494	5.9 M	1498
2018-01-20 18:29:38.320	1797.470	any	10.84.138.200	51(0.9)	887465(5.0)	1.3 G(5.0)	493	5.9 M	1498
2018-01-20 18:29:38.330	1798.760	any	10.195.173.245	52(0.9)	593192(3.3)	888.1 M(3.3)	329	3.9 M	1497
2018-01-20 18:29:38.320	1797.490	any	10.236.120.41	51(0.9)	592418(3.3)	886.9 M(3.3)	329	3.9 M	1497
2018-01-20 18:29:38.320	1798.770	any	10.42.187.65	52(0.9)	592186(3.3)	886.6 M(3.3)	329	3.9 M	1497
2018-01-20 18:29:38.320	1798.800	any	10.163.226.214	51(0.9)	297767(1.7)	445.0 M(1.7)	165	2.0 M	1494
2018-01-20 18:28:39.890	2156.550	any	10.10.82.20	472(8.6)	101576(0.6)	44.0 M(0.2)	47	163224	433
2018-01-20 18:41:51.280	1.180	any	54.239.168.83	2(0.0)	31(0.0)	18288(0.0)	26	123986	589
2018-01-20 18:29:19.240	2113.740	any	10.200.0.2	966(17.5)	23951(0.1)	9.3 M(0.0)	11	35244	388
2018-01-20 18:28:40.510	2152.870	any	10.200.0.6	652(11.8)	27177(0.2)	8.8 M(0.0)	12	32700	323
2018-01-20 18:29:12.300	2084.060	any	10.200.1.2	386(7.0)	23108(0.1)	8.3 M(0.0)	11	31742	357

Summary: total flows: 5519, total bytes: 26621736071, total packets: 17851922, avg bps: 98756758, avg pps: 8278, avg bpp: 1491
Time window: 2018-01-20 18:28:39 - 2018-01-20 19:04:36
Total flows processed: 5519, Blocks skipped: 0, Bytes read: 354784
Sys: 0.010s flows/second: 512156.6 Wall: 0.007s flows/second: 787191.6

Slika 4.5: NFSEN izpis prometne obremenitve R1 in R2.

Iz izpisa lahko hitro ugotovimo, da ima napadena naprava IP naslov 10.200.2.200. Vidni so tudi naslovi napadalčevih tokov in njihova pasovna širina, ki smo jo nastavili na generatorju prometa. S pomočjo orodja Cacti sem spremljal odzivnost strežnika ŽRTEV na ICMP pakete. Po pričakovanjih postane ob napadih strežnik slabo odziven. Delež izgubljenih paketov se veča od prve do zadnje simulacije, saj se veča tudi pasovna širina napada. Tudi po zaustavitvi napada z metodo RTBH se strežnik ŽRTEV še vedno ne oziva na ICMP pakete. saj je strežnik z orodjem Cacti vpet na R1 in v času obrambe napada zavrača vse pakete namenjene strežniku ŽRTEV. Šele ko napad DDOS preneha in odstranimo RTBH usmerjanje, se začne strežnik ŽRTEV ponovno odzivat na ICMP pakete. Slika 4.6 prikazuje neodzivnost napadenega strežnika na ICMP pakete v času izvajanja napadov in času izvajanja metode RTBH in po njej.



Slika 4.6: Odzivnost strežnika ŽRTEV na ICMP promet. Prva simulacija se je izvajala v časih med 14:45 do 17:30 (prvi trije stolpci na grafu), druga simulacija se je izvajala med 17:30 in 19:30 (naslednji trije stolpci na grafu), tretja simulacija pa v času med 19:30 in 20:30 (zadja dva stolpca na grafu). Metoda RTBH je bila sprožena ob času 15:15, 16:00, 17:00, 18:00, 19:00 in 20:16

Uporabljena orodja so torej omogočila hitro detekcijo simuliranega napada. Prav tako je metoda RTBH bila uspešna pri odpravi posledic napada.

5 Zaključek

V diplomskem delu sem se osredotočil na detekcijo in preprečevanje napadov DDOS z uporabo odprtokodnih rešitev in ustreznih omrežnih protokolov. Ta način preprečevanja se uporablja v okoljih manjših ponudnikov internetnih storitev, kjer so komercialne rešitve izven finančnega dosega. Rešitve opisane v diplomskem delu imajo seveda omejitve tako z vidika skalabilnosti kot tudi zanesljivosti. Prva slabost teh rešitev je razvoj, ki je odvisen od aktivnosti odprtokodnih skupnosti. Poleg tega je njihovo delovanje in vzdrževanje popolnoma odvisno od inženirjev zaposlenih pri ponudniku internetnih storitev. Pri plačljivih rešitvah je razvoj, delovanje in vzdrževanje garantirano s strani proizvajalca rešitve. Prikazal sem, da je NFSEN primerno orodje za zaznavo napada DDOS in metoda RTBH učinkovita pri zajezitvi napada. Menim, da sem v diplomskem delu prikazal področje napadov DDOS na ponudnike internetnih storitev, ki ga sicer redko srečamo v akademskih sferah. Mogoče bo moje dipomsko delo vzpodbudilo ponovno razpravo o omenjeni tematiki ter problematike zaščite pred volumetričnimi napadi DDOS.

V praksi se pri reševenju napadov DDOS pogosto uporabljata še dva pristopa. Prvi uporablja princip pregledovanja prometa v živo (ang. *inline*). Ta rešitev v realnem

času pregleduje ves internetni promet ponudnika internetnih storitev in išče anomalije v njem. Njena največja slabost je, da mora namenska strojna oprema slediti potrebam po povečevanju pasovne širine ponudnkika do interneta. Omogoča zelo granularno zaščito. Podobno zaščito bi lahko dosegli tudi s pomočjo *flowspec* BGP začite. Drugi pristop govori o tem, da preko BGP protokola preusmerimo napaden segment omrežja na prečiščevalni center (ang. *scrubbing center*). Tipično je to plačljiva storitev, pri kateri nam ponudnik prečiščevalne storitve preko *GRE* tunela vrne očiščen promet nazaj v omrežje.

Vse rešitve opisane v diplomskem delu imajo svoje prednosti in slabosti. Vedno je potrebno stremeti k rešitvam, ki se najbolj prilegajo našim porebam in čimbolj izpolnjujeno naša pričakovanja.

Literatura

[1] D. Anstee, C. Chui, P. Bowen, G. Sockrider, 12th worldwide infrastructure security report, Tech. rep., Arbor Networks (2017).

[2] K. Hribar, Novi tipi omrežnih napadov in njihovo preprečevanje, Diplomsko delo, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani (2014).

[3] Spletna objava: An introduction to SNMP, https://www.digitalocean.com/community/tutorials/an-introduction-to-snmp-simple-network-management-protocol, [Dostopano: 2. 1. 2018].

[4] Spletna objava, https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.html, [Dostopano: 2. 1. 2018].

[5] Spletna objava: NetFlow, https://en.wikipedia.org/wiki/NetFlow, [Dostopano: 20. 1. 2018].

[6] Spletna objava, https://en.wikipedia.org/wiki/Border_Gateway_Protocol, [Dostopano: 14. 1. 2018].

[7] Spletna objava, http://searchtelecom.techtarget.com/feature/BGP-essentials-The-protocol-that-makes-the-Internet-work, [Dostopano: 14. 1. 2018].

[8] Spletna objava: RRDtool - tutorial and graph examples, https://calomel.org/rrdtool.html, [Dostopano: 2. 1. 2018].

[9] Spletna objava, https://www.cacti.net/what_is_cacti.php, [Dostopano: 2. 1. 2018].

[10] P. Haag, Spletna objava: Watch your flows with NFSEN and NFDump, 50th RIPE Meeting (2005) presentation - http://meetings.ripe.net/ripe-50/presentations/ripe50-plenary-tue-nfsen-nfdump.pdf, [Dostopano: 18. 1. 2018].

[11] Spletna objava: Remotely-triggered black hole (RTBH) routing, http://packetlife.net/blog/2009/jul/6/remotely-triggered-black-hole-rtbh-routing/, [Dostopano: 18. 1. 2018].

[12] C. M. Tim Battles, Danny McPherson, Spletna objava: Customer-triggered real-time blackholes, NANOG 30 (2004) presentation - https://www.nanog.org/meetings/nanog30/presentations/morrow.pdf, [Dostopano: 2. 1. 2018].

[13] Spletna objava: RFC3330 - Special-Use IPv4 Addresses, http://tools.ietf.org/html/rfc3330, [Dostopano: 18. 1. 2018].

[14] Spletna objava: Rfc7674 - clarification of the flowspec redirect extended community, https://tools.ietf.org/html/rfc7674, [Dostopano: 18. 1. 2018].

[15] Spletna objava: ASR9000/XR:Understanding BGP flowspec (BGP-FS), https://supportforums.cisco.com/t5/service-providers-documents/asr9000-xr-understanding-bgp-flowspec-bgp-fs/ta-p/3139916, [Dostopano: 2. 1. 2018].

[16] Spletna objava: VMWARE, https://en.wikipedia.org/wiki/VMware, [Dostopano: 18. 1. 2018].

[17] Spletna objava: RouterOS features, https://wiki.mikrotik.com/wiki/Manual:RouterOS_features, [Dostopano: 18. 1. 2018].

[18] Spletna objava: Install Cacti on CentOS 7– The Definitive Guide in 2017, https://basimaly.wordpress.com/2017/03/02/install-cacti-on-centos-7-the-definitive-guide-in-2017/, [Dostopano: 18. 1. 2018].

[19] Spletna objava: NFSEN - CentOS 7.x, https://wiki.polaire.nl/doku.php?id=nfsen_centos7, [Dostopano: 18. 1. 2018].

[20] Spletna objava: Manual:tools/traffic generator, https://wiki.mikrotik.com/wiki/Manual:Tools/Traffic_Generator, [Dostopano: 18. 1. 2018].

[21] Spletna objava: Manual:queue, https://wiki.mikrotik.com/wiki/Manual:Queue, [Dostopano: 18. 1. 2018].